AMENDMENT TO THE CLAIMS

1-36 (canceled)

37. (currently amended): A method for providing a capability to securely update information stored in a plurality of computer systems, wherein the method comprises:

forming a protected partition within a hard drive of each of the computer systems

storing, within nonvolatile storage of each computer system in the plurality of computer systems, a setup password, an operating system, and an initialization routine to execute within a processor of the computer system after power on of the computer system, wherein the initialization routine includes instructions causing the protected partition to be locked before the operating system is loaded, and wherein instructions causing information stored within ~~the~~ a predetermined location to be written within the protected partition after predetermined security procedures using the setup password have occurred but before the protected partition is locked;

establishing a network connecting each computer system in the plurality of computer systems with a server system;

generating ~~a file~~ an update partition file within the server system;

transmitting the ~~a file~~ update partition file over the network to each computer system in the plurality of computer systems; and

storing the ~~a file~~ update partition file within the predetermined location of each computer system in the plurality of computer systems.

38. (currently amended): The method of claim 37, wherein the initialization routine includes instructions causing the processor of the computer system to perform a method including:

comparing information stored in the ~~update~~ protected partition with information from the ~~a file~~ update partition file stored within the predetermined location;

when a ~~matching~~ portion of the information stored in the protected partition is found match a portion of the information stored within the ~~file~~ update partition file, overwriting the ~~matching~~ portion of the information stored in the protected partition with the portion of the information stored in the ~~file~~ update partition file if space around the ~~matching~~ portion of the information stored in the protected partition is sufficient;

when a portion of the information stored in the protected partition is not found to match a portion of the information stored within the ~~file~~ update partition file, writing the portion of the information stored within the ~~file~~ update partition file to append to the information stored in the protected partition if space within the protected partition is sufficient; and

locking the protected partition to prevent further modification of information stored within the protected partition.

39. (currently amended): The method of claim 38, wherein

a flag bit is set in non-volatile storage within the computing system when the ~~file~~ update partition file is stored at a predetermined location in non-volatile storage within the computing system, and

determining whether the ~~file~~ update partition file is stored within the computing system for updating the protected partition is performed by determining whether the flag bit is set.

40. (currently amended): The method of claim 38, wherein

the method additionally comprises, after determining that the ~~file~~ update partition file is stored within the computing system for updating the protected partition, verifying whether the update partition file has been generated by the server system, and the portion of the ~~file~~ update partition file is written to the protected partition only following verification that the update partition file has been generated by the server system.

41. (currently amended): The method of claim 40, wherein verification that the ~~file~~ update partition file has been generated by the server system includes:

forming a first message digest by applying a hash algorithm to a portion of the ~~file~~ update partition file ;

forming a second message digest by decrypting a digital signature within the ~~file~~ update partition file using a public key of the server system; and;

determining that the first and second message digests are identical.

42. (currently amended): The method of claim 40, wherein

the predetermined setup procedures include verifying that the ~~file~~ update partition file has been generated by the server system includes signing an encrypted portion of the update partition file with a public key of the trusted server system, and

the encrypted portion of the ~~file~~ update partition file has been prepared by signing, with a private key of the server system, a result of the application of an algorithm to data including a version of the setup password accessed by the server system.

43. (currently amended): The method of claim 42, wherein

the data includes the version of the setup password appended to a portion of the ~~file~~ update partition file ,

said algorithm is a hash algorithm generating a message digest, and

verifying that the update partition file has been generated by the server system includes applying the hash algorithm to the setup password stored within the computing system appended to a portion of the update partition file to generate a first version of a message digest and comparing the first version of the message digest with a second version of the message digest obtained by signing the encrypted portion of the update partition file.

44. (currently amended): The method of claim 38, wherein

the ~~file~~ update partition file includes a plurality of entries and a plurality of encrypted elements,

each entry within the plurality of entries includes information to be stored at a different location within the protected ~~file~~ partition,

each encrypted element within the plurality of encrypted elements is associated with an entry in the plurality of entries[[.]],

the method additionally comprises, following determining that the ~~file~~ update partition file is stored within the computing system for updating the protected partition, verifying whether each entry in the plurality of entries within the ~~file~~ update partition file has been generated by the server system, and

each entry in the plurality of entries within the ~~file~~ update partition file is written to the protected partition only following verification that the entry has been generated by the server system.

45. (previously presented): The method of claim 44, wherein verifying that the entry has been generated by the server system includes:

forming a first message digest by applying a hash algorithm to the entry;

forming a second message digest by signing the encrypted element associated with the entry using a public key of the server system; and;

determining that the first and second message digests are identical.

46. (currently amended): The method of claim 44, wherein verifying that the entry has been generated by the server system includes signing the encrypted element associated with the entry with a public key of the server system, and the encrypted element of the ~~file~~ update partition file has been prepared by

signing, with the private key of the server system, a result of the application of an algorithm to data including a version of the setup password accessed by the server system.

47. (previously presented): The method of claim 46, wherein

the data includes the version of the setup password appended to a the entry,

the algorithm is a hash algorithm generating a message digest, and

verifying that the entry has been generated by the server system includes applying the hash algorithm to the setup password stored within the computing system appended the entry to generate a first version of a message digest and comparing the first version of the message digest with a second version of the message digest obtained by signing the encrypted element.

48. (currently amended): The method of claim 44, wherein

information stored in the ~~file update~~ protected partition is compared to each entry in the plurality of entries within the update partition file,

when a portion of the information stored in the protected partition is found to match the entry, the portion of the information stored in the protected partition is overwritten with the entry if space around the ~~matching~~ portion of the information stored in the protected partition is sufficient, and

when a portion of the information stored in the protected partition is not found to match the entry, the entry is appended to the information stored in the protected partition if space within the protected partition is sufficient.

49. (previously presented): The method of claim 38, wherein

the method additionally comprises receiving an input signal from a keyboard of the computing system and comparing the input signal with a signal corresponding to a setup password stored in non-volatile storage within the computing system, and

the protected partition is left unlocked if the input signal matches the signal corresponding to the setup password.

50. (currently amended): An interconnected system for providing updated information in a secure manner, wherein the interconnected system comprises:

a network;

a server system connected to the network and programmed to generate an update partition file and to transmit the update partition file over the network; and

a computer system connected to the network, wherein the computer system includes a processor, non-volatile data storage including a hard drive having a protected partition, wherein the processor is programmed to receive the update partition file from the network and to store the update partition ~~information~~ file in a predetermined location within the nonvolatile data storage outside the protected partition, and wherein the nonvolatile data storage stores an operating system and an initialization routine, executing within the processor after power on of the computer system, including instructions causing the protected partition to be locked before the operating system is loaded, and instructions causing information stored within the predetermined location to be written within the protected partition after predetermined security procedures have occurred but before the protected partition is locked.

51. (currently amended): The ~~method~~ interconnected system of claim 50, wherein the initialization routine includes instructions causing the processor of the computer system to perform a method including:

comparing information stored in the ~~update~~ protected partition with information from the ~~a file~~ update partition file stored within the predetermined location;

when a ~~matching~~ portion of the information stored in the protected partition is found match a portion of the information stored within the update partition file, overwriting the ~~matching~~ portion of the information stored in the protected partition with the portion of the information stored in the protected partition if space around the ~~matching~~ portion of the information stored in the protected partition is sufficient;

when a portion of the information stored in the protected partition is not found to match a portion of the information stored within the update partition file, writing the portion of the information stored within the update partition file to append to the information stored in the protected partition if space within the protected partition is sufficient; and

locking the protected partition to prevent further modification of information stored within the protected partition.

52. (currently amended): The ~~method~~ interconnected system of claim 51, wherein

a flag bit is set in non-volatile storage within the computing system when the ~~file~~ update partition file is stored at a predetermined location in non-volatile storage within the computing system, and

determining whether the ~~file~~ update partition file is stored within the computing system for updating the protected partition is performed by determining whether the flag bit is set.

53. (currently amended): The ~~method~~ interconnected system of claim 51, wherein

the method additionally comprises, after determining that the update partition file is stored within the computing system for updating the protected partition, verifying whether the update partition file has been generated by a trusted server system, and the portion of the update partition file is written to the protected partition only following verification that the update partition file has been generated by the server system.

54. (currently amended): The ~~method~~ interconnected system of claim 53, wherein verification that the update partition file has been generated by the server system includes:

forming a first message digest by applying a hash algorithm to a portion of the update partition file;

forming a second message digest by decrypting a digital signature within the update partition file using a public key of the server system; and;

determining that the first and second message digests are identical.

55. (currently amended): The ~~method~~ interconnected system of claim 53, wherein

the predetermined setup procedures include verifying that the update partition file has been generated by the server system includes signing an encrypted portion of the update partition file with a public key of the trusted server system, and

the encrypted portion of the update partition file has been prepared by signing, with a private key of the server system, a result of the application of an algorithm to data including a version of ~~the~~ a setup password accessed by the server system.

56. (currently amended): The ~~method~~ interconnected system of claim 55, wherein

the data includes the version of the setup password appended to a portion of the update partition file,

the algorithm is a hash algorithm generating a message digest, and verifying that the update partition file has been generated by the server system includes applying the hash algorithm to the setup password stored within the computing system appended to a portion of the update partition file to generate a first version of a message digest and comparing the first version of the message digest with a second version of the message digest obtained by signing the encrypted portion of the update partition file.

57. (currently amended): The ~~method~~ interconnected system of claim 51, wherein

the ~~file~~ update partition file includes a plurality of entries and a plurality of encrypted elements,

each entry within the plurality of entries includes information to be stored at a different location within the protected ~~file~~ partition,

each encrypted element within the plurality of encrypted elements is associated with an entry in the plurality of entries.

the method additionally comprises, following determining that the ~~file~~ update partition file is stored within the computing system for updating the protected partition, verifying whether each entry in the plurality of entries within the ~~file~~ update partition file has been generated by the server system, and

each entry in the plurality of entries within the ~~file~~ update partition file is written to the protected partition only following verification that the entry has been generated by the server system.

58. (currently amended): The ~~method~~ interconnected system of claim 57, wherein verifying that the entry has been generated by the server system includes:

forming a first message digest by applying a hash algorithm to the entry;

forming a second message digest by signing the encrypted element associated with the entry using a public key of the server system; and;

determining that the first and second message digests are identical.

59. (currently amended): The ~~method~~ interconnected system of claim 57, wherein verifying that the entry has been generated by the server system includes signing the encrypted element associated with the entry with a public key of the server system, and the encrypted element of the ~~file~~ update partition file has been prepared by signing, with the private key of the server system, a result of the application of an algorithm to data including a version of ~~the~~ a setup password accessed by the server system.

60. (currently amended): The ~~method~~ interconnected system of claim 59, wherein

the data includes the version of the setup password appended to a the entry,

said algorithm is a hash algorithm generating a message digest, and

verifying that the entry has been generated by the server system includes applying the hash algorithm to the setup password stored within the computing system appended the entry to generate a first version of a message digest and comparing the first version of the message digest with a second version of the message digest obtained by signing the encrypted element.

61. (currently amended): The ~~method~~ interconnected system of claim 57, wherein

information stored in the ~~file update~~ protected partition is compared to each entry in the plurality of entries within the update partition file,

when a portion of the information stored in the protected partition is found to match the entry, the portion of the information stored in the protected partition is overwritten with the entry if space around the ~~matching~~ portion of the information stored in the protected partition is sufficient, and when a portion of the information stored in the protected partition is not found to match the entry, the entry is appended to the information stored in the protected partition if space within the protected partition is sufficient.

62. (currently amended): The ~~method~~ interconnected system of claim 51, wherein

the method additionally comprises receiving an input signal from a keyboard of the computing system and comparing the input signal with a signal corresponding to a setup password stored in non-volatile storage within the computing system, and

the protected partition is left unlocked if the input signal matches the signal corresponding to the setup password.